Amendments to the Claims

Please amend claims to be as follows.

1. (currently amended)   A method of compiling a computer program, the method comprising:

    receiving a plurality of modules of source code;

    generating intermediate representations corresponding to the modules;

    extracting a set of data from the intermediate representations to create an inliner summary for each module; [[and]]

    using the inliner summaries and a globally-sorted working-list based order in an inline analysis phase to determine which call sites in the modules are to be inlined by substituting code from a called module; and

    after a call site is determined to be inlined, updating a call graph of the routines and call sites, and updating the inliner summaries throughout the call graph.

2. (canceled)

3. (currently amended) The method of ~~claim 2~~ claim 1, further comprising, after the call graph and inliner summaries are updated,

    re-calculating profitabilities associated with remaining call sites; and

    re-ordering the working list using the re-calculated profitabilities.

4. (currently amended)   The method of ~~claim 4~~ claim 3, wherein updating the inliner summaries comprises determining nodes and edges of the call graph that are affected by the inlining of the call site and updating those inliner summaries corresponding to the affected nodes and edges.

10/699,144
Amendment and Response to Office Action

Atty. Docket No. 200313003-1
May 16, 2007

5. (currently amended) The method of ~~claim 5~~ claim 4, wherein the edge summaries include at least a call site execution count and a signature type.

6. (currently amended) The method of ~~claim 5~~ claim 4, wherein the node summaries include at least a code size, a routine execution count, and a call-graph height.

7. (original) The method of claim 1, wherein the inline analysis phase is separate and distinct from an inline transformation phase.

8. (currently amended) An apparatus for compiling a computer program, the apparatus comprising:

a processor configured to execute computer-readable code;

a memory system configured to store data;

computer-readable code for a front-end portion of a compiler program, the front-end portion of the compiler program being configured to receive a plurality of modules of source code, generate intermediate representations corresponding to the modules, and extract a set of data from the intermediate representations to generate inliner summaries for the modules; and

computer-readable code for a cross-module optimizer of the compiler program, the cross-module optimizer being configured to use the inliner summaries and a globally-sorted working-list based order to analyze the call sites in an inline analysis phase so as to determine which call sites in the modules are to be inlined by substituting code from a called module, and to update a call graph of the routines and call sites, and to update the inliner summaries, after a call site is determined to be inlined.

9. (canceled)

10/699,144
Amendment and Response to Office Action

Atty. Docket No. 200313003-1
May 16, 2007

10. (currently amended)   The apparatus of ~~claim 10~~ claim 8, wherein the cross-module optimizer is further configured to re-calculate profitabilities associated with remaining call sites, and re-order the working list using the re-calculated profitabilities, after the call graph and inliner summaries are updated.

11. (currently amended)   The apparatus of ~~claim 12~~ claim 10 wherein the cross-module optimizer is further configured to update the inliner summaries by determining nodes and edges of the call graph that are affected by the inlining of the call site and update those inliner summaries corresponding to the affected nodes and edges.

12. (currently amended)   The apparatus of ~~claim 13~~ claim 11, wherein the edge summaries generated by the front-end portion include at least a call site execution count and a signature type.

13. (currently amended)   The apparatus of ~~claim 13~~ claim 11, wherein the node summaries generated by the front-end portion include at least a code size, a routine execution count, and a call-graph height.

14. (currently amended)   The apparatus of ~~claim 1~~ claim 8, wherein the cross-module optimizer is further configured to perform the inline analysis phase separately and distinctly from an inline transformation phase.

15. (currently amended) A computer program product comprising a computer-usable medium having computer-readable code embodied therein, the computer program product being compiled from a plurality of modules of source code using inliner summaries and a globally-sorted working-list based order in an inline analysis phase to

10/699,144
Amendment and Response to Office Action

Atty. Docket No. 200313003-1
May 16, 2007

determine which call sites in the modules are to be inlined by substituting code from a called module <u>and, after a call site is determined to be inlined, to determine a call graph of the routines and call sites, and to update the inliner summaries throughout the call graph</u>.